

UNITED STATES PATENT APPLICATION
FOR

METHOD AND SYSTEM TO ANALYZE
INLINED FUNCTIONS

Inventors:
BRIAN FAHS
ROBERT HUNDT
VINODHA RAMASAMY
TARA KRISHNASWAMY

Prepared by:
WAGNER, MURABITO & HAO LLP
Two North Market Street
Third Floor
San Jose, California 95113

METHOD AND SYSTEM TO ANALYZE INLINED FUNCTIONS

5

TECHNICAL FIELD

10 The present claimed invention relates to analysis of a computer program. More specifically, the present claimed invention relates to analyzing of inlined functions.

BACKGROUND ART

15 Over recent years, the computing community developed a strong set of tools and methods used to analyze and monitor run-time behavior of a program. Performance analysis tools include, for example, basis tools which allow for mapping of periodically taken snapshots during a program's execution to the program's source (e.g. sampling applications)
20 and more complex tools which allow a broader range of program analysis (e.g. code instrumentation applications). Measurements such as basic-block coverage and function invocation counting can be accurately made using code instrumentation. One specific type of code instrumentation is referred to as dynamic binary instrumentation. Dynamic binary
25 instrumentation allows program instructions to be changed on-the-fly. Additionally, dynamic binary instrumentation, as opposed to static instrumentation, is performed at run-time of a program and only instruments those parts of an executable that are actually executed. This minimizes the overhead imposed by the instrumentation process itself.
30 Furthermore, performance analysis tools based on dynamic binary instrumentation require no special preparation of an executable such as, for example, a modified build or link process.

 Unfortunately the benefits of conventional performance analysis
35 tools are not available to all types of programs and functions. Specifically, conventional performance analysis tools will not work properly with inlined functions. As an explanation, many programming languages offer support for "inlining" functions. That is, many programming languages such as, for example, C++, allow the compiler to generate
40 machine code for a function call such that the code from the function body

gets directly inserted into the place where the call was made. The now inlined function causes the size of the text program to increase but removes the overhead of the function call. From the point of view of the programmer, there is some ambiguity as to whether a particular function has been inlined or not. For example, even if the programmer specifies in the source code that a certain function be inlined, that does not necessarily mean that the particular function will be ultimately be inlined in the binary executable by the compiler. This ambiguity exists because there are certain cases where the compiler decides, on its own, not to inline a function even though the programmer has specified for the function to be inlined. Because conventional performance analysis tools correlate to the binary executable and the regular functions therein as opposed to the source code, and because conventional performance analysis tools do not take into account inlined function information, inlined functions can not be properly analyzed using existing performance analysis tools.

Thus, a need has arisen for a method and system for examining an inlined function using a performance analysis tool.

DISCLOSURE OF THE INVENTION

The present invention provides a method and system for examining an inlined function using a performance analysis tool.

5 Specifically, in one method embodiment, the present invention identifies an inlined function. Upon identification of the inlined function, the present embodiment uses a performance analysis tool to perform a desired task on the inlined function.

10 These and other technical advantages of the present invention will no doubt become obvious to those of ordinary skill in the art after having read the following detailed description of the preferred embodiments which are illustrated in the various drawing figures.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention:

FIGURE 1 is a schematic diagram of an exemplary computer system used to perform steps of the present method in accordance with various embodiments of the present claimed invention.

FIGURE 2 is a flow chart of steps performed to analyze an inlined function in accordance with one embodiment of the present claimed invention.

FIGURE 3 is a flow chart of steps performed to analyze an inlined function including the step of creating a data structure in accordance with one embodiment of the present claimed invention.

The drawings referred to in this description should be understood as not being drawn to scale except if specifically noted.

BEST MODES FOR CARRYING OUT THE INVENTION

Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. While the invention will be described in conjunction with the preferred embodiments, it will be understood that they are not intended to limit the invention to these embodiments. On the contrary, the invention is intended to cover alternatives, modifications and equivalents, which may be included within the spirit and scope of the invention as defined by the appended claims. Furthermore, in the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be obvious to one of ordinary skill in the art that the present invention may be practiced without these specific details. In other instances, well known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects of the present invention.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present invention, discussions utilizing terms such as "identifying", "creating", "using", "performing", or the like, refer to the actions and processes of a computer system, or similar electronic computing device. The computer system or similar electronic computing device manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission, or display devices. The present invention is also well suited to the use of other computer systems such as, for example, optical and mechanical computers.

COMPUTER SYSTEM ENVIRONMENT OF THE PRESENT INVENTION

With reference now to Figure 1, portions of the present method and system are comprised of computer-readable and computer-executable instructions which reside, for example, in computer-usable media of a computer system. Figure 1 illustrates an exemplary computer system used in accordance with one embodiment of the present invention. It is

appreciated that system 100 of Figure 1 is exemplary only and that the present invention can operate on or within a number of different computer systems including general purpose networked computer systems, embedded computer systems, routers, switches, server devices, client devices, various intermediate devices/nodes, stand alone computer systems, and the like. Additionally, computer system 100 of Figure 1 is well adapted having computer readable media such as, for example, a floppy disk, a compact disc, and the like coupled thereto. Such computer readable media is not shown coupled to computer system 100 in Figure 1 for purposes of clarity. Additionally, portions of the present embodiment are well suited to operating in conjunction with various mobile clients such as, for example, a cell phone, personal digital assistant (PDA), laptop computer, pager, and the like.

System 100 of Figure 1 includes an address/data bus 102 for communicating information, and a central processor unit 104 coupled to bus 102 for processing information and instructions. As an example, central processor unit 104 may be an IA-64 microprocessor architecture by Intel Corporation of Santa Clara, California. System 100 also includes data storage features such as a computer usable volatile memory 106, e.g. random access memory (RAM), coupled to bus 102 for storing information and instructions for central processor unit 104. System 100 also includes computer usable non-volatile memory 108, e.g. read only memory (ROM), coupled to bus 102 for storing static information and instructions for the central processor unit 104. Such static information is comprised, in one embodiment, of commands for configuration and initial operations of computer system 100. Computer system 100 also includes a data storage unit 110 (e.g., a magnetic or optical disk and disk drive) coupled to bus 102 for storing information and instructions.

30

System 100 of the present invention also includes an optional alphanumeric input device 112 including alphanumeric and function keys coupled to bus 102 for communicating information and command selections to central processor unit 104. System 100 also optionally includes an optional cursor control device 114 coupled to bus 102 for communicating user input information and command selections to central processor unit 104. System 100 of the present embodiment also includes an optional display device 116 coupled to bus 102 for displaying information. System 100 of the present embodiment also includes a

communication interface 118 which enables computer system 100 to interface with other computers or devices. In one embodiment, communication 118 is, for example, a modem, an integrated services digital network (ISDN) card or the like, a local area network (LAN) port, etc. Those skilled in the art will recognize that modems or various types of network interface cards (NICs) typically provide data communications via telephone lines, while a LAN port provides data communications via a LAN. Communication interface 118 of computer system 100 may also enable wireless communications. Furthermore, communication interface 118 may enable communication with other computers or devices through one or more networks. For example, computer system 100, using communication interface 118, may communicate to the "Internet."

Computer system 100 may be used to implement the techniques described below. In various embodiments, processor 104 performs the steps of the techniques by executing instructions brought to RAM 106. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the described techniques. Consequently, embodiments of the invention are not limited to any one or a combination of software, hardware, or circuitry.

Instructions executed by processor 104 may be stored in and carried through one or more computer-readable media, which refer to any medium from which a computer reads information. Computer-readable media may be, for example, a floppy disk, a hard disk, a zip-drive cartridge, a magnetic tape, or any other magnetic medium, a CD-ROM, a CD-RAM, a DVD-ROM, a DVD-RAM, or any other optical medium, paper-tape, punch-cards, or any other physical medium having patterns of holes, a RAM, a ROM, an EPROM, or any other memory chip or cartridge. Computer-readable media may also be coaxial cables, copper wire, fiber optics, acoustic, or light waves, etc. As an example, the instructions to be executed by processor 104 are in the form of one or more software programs and are initially stored in a CD-ROM being interfaced with computer system 100. Computer system 100 loads these instructions in RAM 106, executes some instructions, and sends some instructions via communication interface 118, a modem, and a telephone line to a network, the Internet, etc. A remote computer, receiving data through a network cable, executes the received instructions and sends the data to computer system 100 to be stored in storage device 110.

Referring still to Figure 1, optional display device 116 of Figure 1, may be a liquid crystal device, cathode ray tube, or other display device suitable for creating graphic images and alphanumeric characters recognizable to a user. Optional cursor control device 114 allows the computer user to dynamically signal the two dimensional movement of a visible symbol (cursor) on a display screen of display device 116. Many implementations of cursor control device 114 are known in the art including a trackball, mouse, touch pad, joystick or special keys on alphanumeric input device 112 capable of signaling movement of a given direction or manner of displacement. Alternatively, it will be appreciated that a cursor can be directed and/or activated via input from alphanumeric input device 112 using special keys and key sequence commands. The present invention is also well suited to directing a cursor by other means such as, for example, voice commands. A more detailed discussion of the present invention is found below.

GENERAL METHOD AND SYSTEM FOR ANALYZING AN INLINED FUNCTION

With reference next to flow chart 200 of Figure 2 and to Figure 1, exemplary steps used by the various embodiments of present invention are illustrated. Flow chart 200 includes processes of the present invention which, in one embodiment, are carried out by a processor under the control of computer-readable and computer-executable instructions. The computer-readable and computer-executable instructions reside, for example, in data storage features such as computer usable volatile memory 106, computer usable non-volatile memory 108, and/or data storage device 110 of Figure 1. In one embodiment, the computer-readable and computer-executable instructions are used to control or operate in conjunction with, for example, processor 104 of Figure 1.

With reference again to Figure 2, steps performed in accordance with one embodiment of the present invention are shown. Although specific steps are disclosed in flow chart 200 of Figure 2, such steps are exemplary. That is, the present invention is well suited to performing various other steps or variations of the steps recited in Figure 2. At step 202, the present embodiment identifies an inlined function. In one embodiment (e.g. a C++ application environment), the present embodiment browses through the debug information present in the target application and obtains location information for the inlined function.

Specifically, in this embodiment, the present invention obtains and reads the source correlation information contained within the executable. The present embodiment uses the source correlation information to obtain the start and end addresses for the inlined function. The example set forth

5 herein has and will continue to refer to an inline function (i.e. one inlined function). It should be understood, however, that such an example is for purposes of illustration only, and that the present invention is also well suited to an embodiment in which the present invention performs the steps of the present example for a plurality of inlined functions.

10

Referring now to step 204, the present embodiment then uses a performance analysis tool to perform a desired task on the inlined function. More specifically, in one embodiment, using the start address obtained at step 202 for the inlined function, the present invention inserts

15 a breakpoint or other control transfer instruction at the start of the inlined function. Additional details related to instrumentation including discussion of features such as breakpoints can be found in co-owned, commonly-assigned U.S. Patent Application Serial No. 09/833,248 filed April 11, 2001, entitled "Dynamic Instrumentation Of An Executable

20 Program", to Hundt et al. which is incorporated herein by reference as background material. When the breakpoint associated with an inlined function is executed, the present embodiment will perform the desired task on the inlined function.

25

With reference still to step 204, in one embodiment, when the breakpoint associated with an inlined function is executed, the present embodiment will perform an instrumentation task on the inlined function. In such an embodiment, an instrumenting application is used to perform the instrumentation task. In one approach, the instrumenting

30 application is comprised of the Caliper application by Hewlett-Packard Company of Palo Alto, California. The present invention is, however, well suited to use with various other instrumenting applications. In one such embodiment, the instrumenting application will perform the desired instrumentation task (e.g. incrementing a function count), restore the

35 original instructions, if necessary and then continue execution.

With reference still to step 204, in one embodiment when an inserted breakpoint is hit, the instrumenting application will relocate the instrumented function to shared memory and replace the function entry

with a long branch to shared memory or memory of the target process. In such an approach, the instrumenting application inserts code into the relocated instrumented function to, for example, increment counters placed at the start of each inlined function. Additionally, another
 5 approach besides relocating the instrumented function is replacing the original starting bundle of each inlined function with a long branch to a shared memory probe code sequence that will first save the necessary registers, execute the original bundle, and increment a variable (or perform some other task), restore the registers and then jump back to the
 10 original function.

With reference still to step 204, the present invention is also well suited to an embodiment in which the performance analysis task is comprised, for example, of a task other than an instrumentation task. As
 15 an example, in one embodiment, the present invention performs a mapping of samples to the inlined function. That is, the performance analysis tool is comprised of a sampling application which maps samples to the inlined function located at step 202. Also, in one embodiment, processor 104 of Figure 1 in conjunction with instructions, residing, for
 20 example, in RAM 106, ROM 108, and/or data storage device 110, comprise an apparatus which operates to perform steps 202 and 204.

With reference now to Figure 3, a flow chart 300 of steps performed in accordance with another embodiment of the present invention is
 25 shown. The method of the present embodiment includes steps 202 and 204 of Figure 2. These steps were described above in detail in conjunction with the description of Figure 2, and are not repeated herein for purposes of brevity and clarity. The method of the present embodiment as described in Figure 3 also includes new step 302. At step 302, the present embodiment,
 30 creates a data structure which maintains location information for the inlined function and information related to the desired task for the inlined function. In one such approach, the data structure resides, for example, in shared memory, maintains the start and end addresses for the inlined function, and also stores information regarding the desired task to be
 35 performed on the inlined function. Information regarding the task to be performed on the inlined function includes, for example, the specific performance analysis task to be performed, results of the performance analysis task, and the like. In one embodiment, processor 104 of Figure 1 in conjunction with instructions, residing, for example, in RAM 106,

ROM 108, and/or data storage device 110, comprise an apparatus which operates to perform step 302.

Thus, the present invention provides a method and system for
5 examining an inlined function using a performance analysis tool.

The foregoing descriptions of specific embodiments of the present invention have been presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the
10 precise forms disclosed, and obviously many modifications and variations are possible in light of the above teaching. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications
15 as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the Claims appended hereto and their equivalents.